

A Fast Carrier Scheduling Algorithm for Battery-free Sensor Tags in Commodity Wireless Networks

Carlos Pérez-Penichet*, Dilushi Piumwardane*, Christian Rohner* and Thiemo Voigt*,†

*Uppsala University, Sweden. †RISE SICS, Sweden.

Email: {carlos.penichet, dilushi.piumwardane, christian.rohner, thiemo.voigt}@it.uu.se

Abstract—New battery-free sensor tags that interoperate with unmodified standard IoT devices and protocols can extend a sensor network’s capabilities in a scalable and cost-effective manner. The tags achieve battery-free operation through backscatter-related techniques, while the standard IoT devices avoid additional dedicated infrastructure by providing the unmodulated carrier that tags need to communicate. However, this approach requires coordination between devices transmitting, receiving and generating carrier, adds extra latency and energy consumption to already constrained devices, and increases interference and contention in the shared spectrum. We present a scheduling mechanism that optimizes the use of carrier generators, minimizing any disruptions to the regular nodes. We employ time slots to coordinate the unmodulated carrier while minimizing latency, energy consumption and overhead radio emissions. We propose an efficient scheduling algorithm that parallelizes communications with battery-free tags when possible and shares carriers among multiple tags concurrently. In our evaluation we demonstrate the feasibility and reliability of our approach in testbed experiments. We find that we can significantly reduce the excess latency and energy consumption caused by the addition of sensor tags when compared to sequential interrogation. We show that the gains tend to improve with the network size and that our solution is close to optimal on average.

I. INTRODUCTION

Backscatter communication and related techniques enable new devices that can perform bidirectional communications compatible with commodity IoT radios when assisted by an external unmodulated carrier [1]–[4]. This class of devices is attractive because of two reasons: First, their dramatically reduced power consumption, compared to regular low-power radios, allows them to leverage the unmodulated carrier or a wide variety of other energy harvesting techniques to operate without batteries. Second, because they are directly compatible with commodity IoT transceivers, they can facilitate deployment and maintenance in novel applications with sensors embedded in the infrastructure [5], medical implants or wearable devices [6]; where having batteries in all nodes may be impractical. This class of battery-free devices is also characterized by a short communication range, generally seen as their main disadvantage.

Scenario. We have previously proposed that devices with these characteristics, called *sensor tags*, could augment a previously-deployed sensor network [2], [4]. By placing sensor tags next to regular nodes, not unlike installing new wireless peripherals to our computers, one can provide them with new sensing

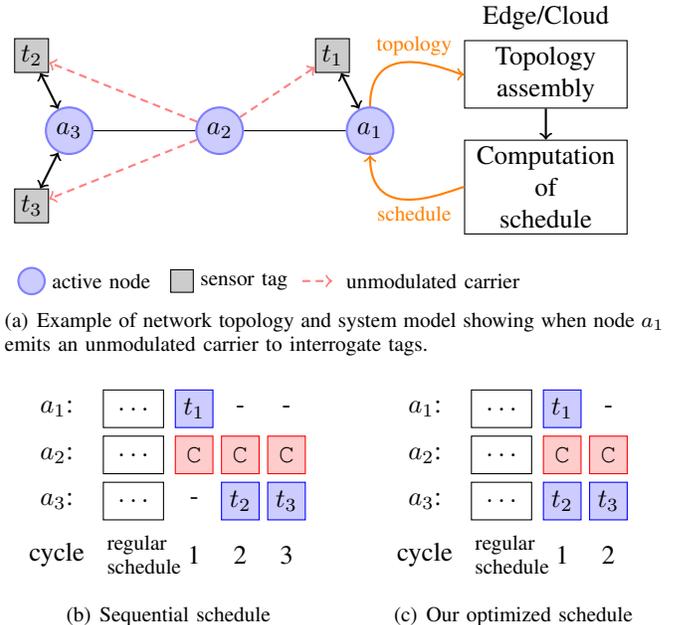


Fig. 1. Our schedule guarantees the timely availability of the unmodulated carrier that sensor tags need to communicate. We optimize the schedule by sharing unmodulated carriers for multiple tags and interrogating them in parallel. This reduces latency, power consumption and spectrum usage compared to the sequential schedule.

capabilities without any physical modifications, and without the extensive maintenance and deployment costs of adding new battery-powered devices. Furthermore, sensor tags enable installing sensors in hard to reach locations, while more capable devices are placed nearby where batteries can be replaced with ease [6]. In these applications the regular nodes can interrogate the tags for sensor readings, coordinating among themselves to wake up and provide the unmodulated carrier that tags need to communicate. This way regular nodes avoid the need for dedicated carrier generators, while also performing their own sensing and providing services like routing, computation, and edge/cloud access. To make these applications practical the already-constrained regular nodes must ensure efficient use of energy and spectrum when interrogating sensor tags.

Contribution. In this paper, we focus on the problem of efficiently scheduling unmodulated carriers in a scenario like

the one on Figure 1(a), where regular nodes cooperate to interrogate a set of sensor tags. This problem differs from the usual scheduling in wireless networks in that, besides coordinating transmitters and receivers, we must also account for carrier generators. The objective is to minimize the amount of resources that the regular nodes need to invest in these interrogations in terms of energy consumption and spectrum usage, as illustrated on Figures 1(b) and 1(c). The key idea is that we leverage the short communication range of sensor tags (Figure 2) to enable spatial reuse to communicate with multiple sensor tags concurrently, often sharing carrier generators. Our scheme limits the overhead energy consumption, spectrum usage and latency resulting from adding tags to the network. We make the following specific contributions:

- We propose an efficient approximate algorithm that optimizes carrier allocation and scheduling to support multiple tag interrogations at the same time, possibly sharing carrier generators.
- Through testbed experiments we demonstrate the feasibility and reliability of our schedule, proving that it limits the disruption caused to regular nodes in terms of overhead energy consumption, spectral usage and communication latency without impacting reliability.
- A systematic numerical analysis evaluates the approximation ratio and bounds of our solution at a larger scale.

Challenges. To avoid the need for dedicated devices to provide the unmodulated carrier, it is desirable to make the regular nodes provide it. This type of carrier support, however, requires coordination between the tag and the regular nodes taking the roles of carrier generator and interrogator. Efficient coordination involves careful network-wide orchestration to parallelize interrogations while avoiding collisions and avoid wasting valuable battery power and radio spectrum, or violating radio emission regulations. The challenge is to compute an efficient schedule and carrier allocation that reuses carrier generators as much as possible and does not produce unnecessary overhead. This leads to an NP-hard combinatorial optimization problem. Therefore, to provide a scalable solution, we must devise an efficient approximate scheduling algorithm and we must ensure that its solutions are close to the optimal ones.

Approach. To guarantee coordination between transmitters, receivers and carrier generators, we employ a time-slotted protocol to pre-assign the function of every device during every time slot. Communications among regular nodes are scheduled independently, using existing TDMA scheduling mechanisms. Additional slots are then appended to the original schedule to contain the tag interrogation schedule (c.f., Figure 1(b)).

We approach slot assignment in the system as a combinatorial optimization problem, which we show is NP-hard. We propose an efficient algorithm that takes the network topology as input and computes a schedule that reduces the time and number of carrier generator slots required to interrogate every tag in the network without collisions, as illustrated in Figure 1.

We employ a testbed to show the feasibility and robustness of our design. Multiple testbed topologies help us study the

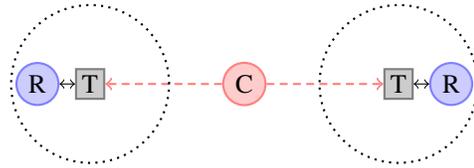


Fig. 2. The interference range of tags (T) depends on the distance to the carrier generator (C). We leverage this property to limit the tags' collision domain to a small region next to the receiver (R) and increase spatial reuse.

properties of our problem in larger realistic scenarios. Finally, additional synthetic topologies help us increase the scale of our evaluation and explore some of its bounds.

Results. We show that our algorithm can be implemented in polynomial time and with sublinear approximation ratio. Our testbed experiments show the feasibility of our solution, guaranteeing the availability of the unmodulated carrier when needed, without impacting reliability. Our schedule guarantees that the carrier is provided with minimal energy and spectrum overhead. Our numerical evaluation shows that we can decrease the added latency by up to a factor that is inversely proportional to the network size, when compared to interrogating every tag in sequence, which requires one dedicated slot and carrier generator each. We also reduce the energy spent per tag interrogation by up to the same factor, depending on the density of the network topology. Finally, our approximate solution is close to the optimal and it appears to grow sub-linearly with the size of the network and with the density of deployed tags.

II. BACKGROUND

Battery-free tags depend on an external unmodulated carrier both for transmission and for reception. Offloading the carrier, one of the most energy intensive elements of a transceiver, to an external device is the key enabler for ultra-low power consumption. To transmit, the device employs backscatter communication techniques that selectively reflect an external Radio Frequency (RF) signal to modulate it and convey information [1], [7] while achieving a power consumption up to three orders of magnitude lower than traditional radios. Multiple examples with WiFi [1], [8], Bluetooth [9], IEEE 802.15.4 [1], [2] and LoRa [10]–[12] have recently become available in the literature. An external RF signal can help a receiver operate with ultra-low power consumption in a way analogous to backscatter [3], [4].

Carrier self-interference. A crucial aspect when using this type of communications is that the unmodulated carrier could interfere with the receiver. To avoid this, the carrier is placed at a different channel than the signal of interest [6], [8]. Sensor tags leverage this principle to avoid interference from carrier-generating nodes.

Communication range and interference. Communication range in battery-free devices depends on the strength of the external carrier [1], [4], [13]. Because we operate in the bistatic case, where the carrier generator and receiver are not co-located, the farther the device is to the carrier generator the

shorter the communication range is and, as a consequence, the smaller its collision domain becomes. Figure 2 illustrates that when the carrier generator is far away from the tag, its communication range and collision domain are small. We leverage this short communication range to limit the collision domain of tag communications to a small region around the tag and enable spatial reuse. Tags cannot operate properly while provided with multiple unmodulated carriers, as the random phase and frequency offsets among the carriers would cause problems for both transmission and reception. Therefore there is always the restriction in our schedule computation that tags must receive a single unmodulated carrier to communicate.

III. SYSTEM MODEL

We focus on the problem of scheduling tag interrogations in a heterogeneous network, like the one in Figure 1(a), consisting of a set of N_a active nodes $A = \{a_1, \dots, a_{N_a}\}$ and a set of N_t tags $T = \{t_1, \dots, t_{N_t}\}$. Active nodes are standard devices with radio transceivers that support commodity physical layer protocols such as Bluetooth or IEEE 802.15.4 and have a means to generate an unmodulated carrier at a chosen frequency, for instance using their radio test mode [2] or other means [6]. The network of active nodes runs on a TDMA Medium Access Control (MAC) protocol with a schedule computed independently and that consists of a set of time slots of duration T_s that together form a slotframe, which cyclically repeats in time. At least one of the active nodes has a connection to an Edge/Cloud server where we compute the tag interrogation schedule. The schedule determines the function that every node should perform during every time slot. We represent the schedule as a map $S_{a,s} : a \in A, s \in [1, N_t] \mapsto T \cup \{C, 0\}$ that indicates the function that node a should perform during cycle s , whether to remain off (0), emit a carrier (C) or interrogate the indicated tag from T (Figure 4).

Sensor tags are equipped with ultra-low-power transceivers with all the characteristics described in Section II, including the need for an external carrier at a shifted frequency and limited communication range depending on carrier strength. Every tag is located next to an active node, said to be its *host*, within a radius much smaller than the typical inter-node distance of the active nodes. An active node may host multiple sensor tags. We model the tag-to-host assignment with a mapping $H_t : t \in T \mapsto n \in A$ that is known a priori in the cloud.

As a consequence of their low-power design, tags can only perform very simple operations such as replying to an interrogation after a short time interval, ex. similar to a regular RFID tag. We do not assume that tags are able to maintain synchrony with the network of regular nodes as they may be intermittently powered like RFID tags.

We represent the network topology as an undirected graph $G = (A, E)$ like the ones in Figure 4, where the vertex set is the set of active nodes A and the edge set E models radio links among active devices. The weighted adjacency matrix $W_{i,j}$ represent the received signal strength observed at node i coming from node j . We assume that the signal strength

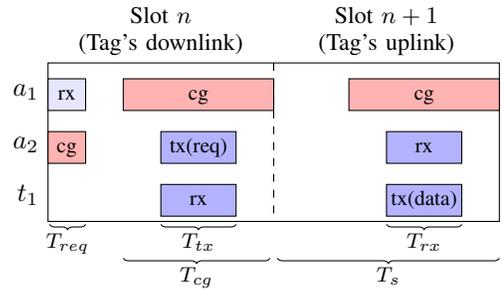


Fig. 3. Spreading interrogation cycles over two timeslots prevents disruption to regular nodes. The interrogating active node (a_2) requests the carrier generator (a_1) to engage by sending a short carrier of its own (cg). With the carrier enabled, a_2 sends a request to the tag (t_1). After a certain delay, the tag replies during the next time slot.

observed at tags is roughly the same as the one observed at its host due to their physical proximity.

IV. DESIGN

The goal of our design is to guarantee that the unmodulated carrier is provided efficiently for all tags whenever necessary. Our objective is to minimize the time that regular nodes must invest in tag interrogations and performing carrier generation; looking to reduce the energy consumption, save spectrum and avoid disruptions to the operation of the regular network.

During bootstrap the cloud server collects information to build the topology graph G . We use the graph as input to compute an optimized schedule which we then disseminate to the nodes, where it is appended to the end of the slotframe containing the regular schedule. During runtime, the cloud may continue to gather information to update the graph and recompute the schedule if necessary to account for changes in the topology and varying link state.

For simplicity we spread interrogation cycles across two consecutive timeslots, as illustrated in Figure 3. The tags' downlink communications happen in one slot and the uplink in the slot immediately after. This enables tags to support full-length frames in both directions without disrupting the timeslot duration for the regular nodes, therefore preserves compatibility with the active nodes' original MAC.

Figure 3 shows the procedure that nodes follow to interrogate a sensor tag. At the beginning of the interrogation cycle, the node assigned to generate carrier (a_1) listens for a period T_{req} . If a request arrives, in the form of another unmodulated carrier, the carrier engages for a duration T_{cg} during both the current and subsequent slot. The interrogating node (a_2) then transmits its request addressed to the desired tag (t_1), which will receive it due to the unmodulated carrier being provided. In the next slot, the carrier engages again so that the tag can transmit the response.

A. Optimized Carrier Generator Scheduling

We now focus on the computation of the schedule. We first present the problem and prove that it is NP-hard. We then propose an approximate algorithm that enables efficient

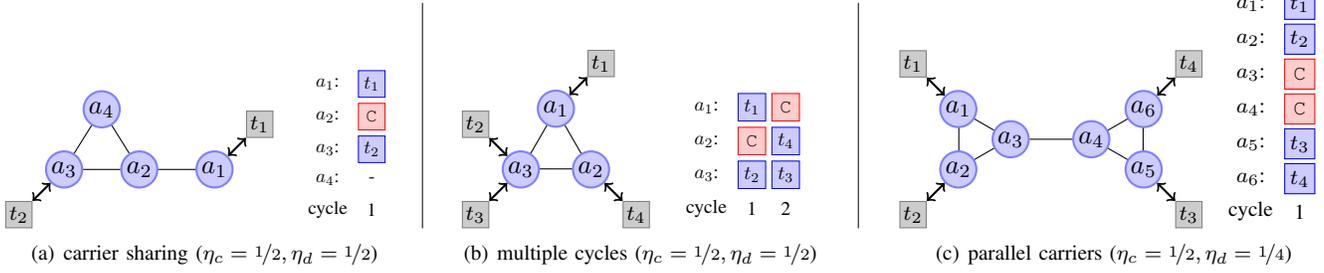


Fig. 4. Example topologies and corresponding TagAlong schedules. The examples highlight different aspects of the scheduling process such as opportunities for carrier sharing and parallel carriers.

computation of the schedule. We analyze its run time and overhead introduced by tag interrogations.

The straightforward way of scheduling sensor tag interrogations is to sequentially assign a dedicated slot and carrier generator to each tag in the network, as shown in Figure 1(b). This approach, however, leads to unnecessarily long schedules that result in excessive latency and wasted energy and spectrum. Intuitively we see that we can leverage the tags' short communication range to make important energy and spectrum savings by interrogating multiple tags concurrently without collisions, often using the same carrier for several tags as in examples of Figure 4.

Problem formulation. The objective in our optimization problem is to find a slot assignment so that all battery-free tags in the network can be interrogated once, with the lowest number of carrier generator slots and in the shortest time, without carrier collisions. This can be expressed formally as:

Problem 1. Tag scheduling.

$$\min (m, |\{s_{i,j} \in S : s_{i,j} = C, \forall i \in A, j \in [1,m]\}|) \quad (1)$$

$$\text{s.t.: } \forall t \in T \exists! c \in [1,m] S_{H_t,c} = t \quad (2)$$

$$\forall t \in T, c \in [1,m] S_{H_t,c} = t \Rightarrow \exists! g \in A \{H_t, g\} \in E \wedge W_{H_t,g} \geq w_{min} \quad (3)$$

Constraint 2 enforces that every tag is interrogated exactly once by its host, while constraint 3 ensures exactly one sufficiently-strong carrier per tag. The objective (1) is to minimize m , the duration of the schedule in interrogation cycles and the total number of times carriers are scheduled.

1) *NP-hardness:* We now show that Problem 1 is NP-hard by restricting it to Minimum Set Cover, a well-known NP-hard problem: Given a collection M of subsets of a set N , find a subset $M' \subseteq M$ such that every element of N belongs to at least one element of M' . The objective is to minimize $|M'|$.

Theorem 1. Tag scheduling is NP-hard.

Proof: We restrict Problem 1 to the Minimum Set Cover problem by allowing only instances like the one shown in

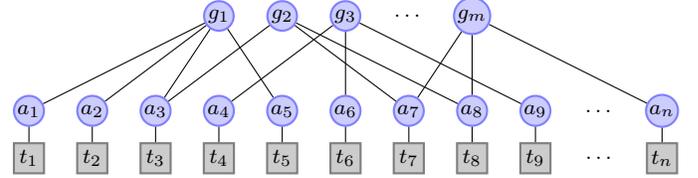


Fig. 5. We restrict our problem to the Minimum Set Cover problem by considering instances like this. The solutions to Minimum Set Cover correspond to the set of enabled carrier generators in Problem 1.

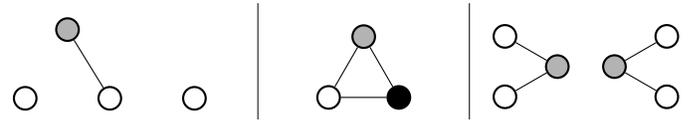


Fig. 6. Conflict graph and colorings corresponding to the example topologies of Figure 4.

Figure 5, where:

$$A = \{g_i : i \in M\} \cup \{a_i : i \in N\}$$

$$E = \{\{i, j\} : i \in M, j \in M_i\}$$

$$T = \{t_i : i \in N\}$$

$$H_{t_i} = a_i, \forall i \in N$$

$$w_{min} = 0$$

The idea is to have two sets of nodes. A group of nodes (a_i) that host one tag each, and correspond to the elements of set N . Another group of nodes (g_i) hosts no tags and corresponds to the elements of collection M . We add edges between nodes a_i and g_j if element j of collection M contains element i of set N . A solution for Problem 1, would select the minimum-cardinality set $\{g_i : i \in M'\}$ of carrier generators needed to interrogate all tag-bearing nodes. This set is also a solution to the Minimum Set Cover problem. Therefore Problem 1 is NP-hard given that a polynomial time solution would also be a solution to the Minimum Set Cover problem. ■

2) *Approximate algorithm:* Because Problem 1 (Tag scheduling) is NP-hard, we propose an approximate scheduling algorithm that we describe next.

Our approximate algorithm, described in Algorithm 1, takes a greedy approach; scheduling as many tags as possible, one timeslot at a time, until all tags are scheduled. The core idea

Algorithm 1 Tag scheduling algorithm

```
1: function SCHEDULE( $G(A, E), W, T, H, w_{min}$ )
2:    $S \leftarrow$  new empty schedule
3:   while  $T \neq \emptyset$  do     $\triangleright$  While unscheduled tags remain
4:      $\forall_{a \in A} s(a) \leftarrow 0$      $\triangleright$  Initialize new all-off slot  $s$ 
5:      $G' \leftarrow$  CONFLICT_GRAPH( $G$ )
6:      $C \leftarrow$  COLOR_GRAPH( $G'$ )
7:     sort  $C$  in descending order of tags served
8:     for all  $c \in C$  do
9:       for all  $g \in A : C(g) = c \wedge s(g) = 0$  do
10:        if  $\neg \exists_{n \in A} \{n, g\} \in E \wedge s(n) \notin \{0, C\}$  then
11:          for all  $n : \{n, g\} \in E$  do
12:            if  $\exists_{t \in T} H_t = n \wedge W_{n,g} > w_{min}$  then
13:               $s(g) \leftarrow C$ 
14:               $s(n) \leftarrow t$ 
15:            end if
16:          end for
17:        end if
18:      end for
19:    end for
20:     $S \leftarrow S + s$      $\triangleright$  Append new slot to schedule
21:     $T' \leftarrow$  tags scheduled in  $s$ 
22:    if  $T' = \emptyset$  then
23:      return None     $\triangleright$  Unsatisfiable
24:    end if
25:     $T \leftarrow T \setminus T'$      $\triangleright$  Remove scheduled tags
26:  end while
27:  return  $S$ 
28: end function
```

is to guarantee that when a tag is interrogated, exactly one neighbor is generating a carrier.

The algorithm computes a *conflict graph* $G'(A, E')$ for graph $G(A, E)$ (line 5). Graphs G and G' share the same set of vertices. The set of edges of G' is such that an edge exists between nodes i and j iff they have at least one common neighbor in graph G with associated tags. This means that i and j should not generate a carrier simultaneously or the carriers would interfere at their common neighbors. A graph coloring of graph G' (line 6) identifies subsets of nodes that can generate carriers simultaneously without causing problems for any tag. Figure 6 shows the conflict graphs and colorings for the example topologies of Figure 4. We then iterate over the set of colors in descending order of the number of tags that would be served by each color c (lines 7 and 8). For each potential carrier generator g colored c , we verify that it has not been assigned any task in the current timeslot (line 9) and that it would not cause interference for any of its already-scheduled neighbors (line 10). For every tag-bearing neighbor n of node g that overcomes the signal strength threshold w_{min} (line 12), we schedule node g as carrier generator for node n to interrogate one of its tags. After testing all colors, the resulting scheduled timeslot is appended to the main schedule (line 20), any scheduled tags are removed from the network (line 25) and the procedure is repeated for a new

timeslot. It will finish when all tags are scheduled or when it finds a slot where it is unable to schedule any tags (line 23), in which case the instance is unsatisfiable because there is no sufficiently strong carrier generator for one or more tags.

3) *Time complexity*: A high-level analysis of Algorithm 1 allows us to determine its time complexity. The loop of line 3 repeats at most N_t times. The construction of the conflict graph in line 5 iterates over every neighbor of every node, therefore, it computes in time $\mathcal{O}(N_a^2)$. To color the graph in line 6, we employ a “largest first” greedy algorithm [14] that runs in $\mathcal{O}(N_a + |E|)$ time and has a coloring performance guarantee of $\mathcal{O}(N_a)$. The sort operation in line 7 must compute the number of tags served by every potential carrier generator, therefore it takes $\mathcal{O}(N_a^2)$ time. The two loops in lines 8 and 9 combine to iterate a total of N_a times. Finally, the loop of line 11 repeats N_a times in the worst case. The set operations of lines 20 to 25 can be implemented in linear time. The worst-case time complexity of Algorithm 1 is therefore $\mathcal{O}(N_t N_a^2)$.

B. Overhead analysis

We now derive analytical expressions to describe the cost of interrogating sensor tags with our schedule, compared to the sequential one, in terms of latency and energy consumption.

1) *Energy overhead*: We define the *carrier ratio* as the fraction of carrier generation cycles in our solution (n_c) relative to the number of cycles needed to sequentially interrogating all tags (N_t cycles) ($\eta_c = n_c/N_t$). By definition $\eta_c = 1$ for the sequential schedule and is related to the energy overhead.

The total energy per slotframe and per tag that active nodes invest to interrogate all tags is given by $\bar{E} = \bar{E}_{tx} + \bar{E}_{rx} + \bar{E}_{cg}$. Where the terms on the right are the energy invested in transmissions, reception and carrier generation respectively. From Figure 3 it is easy to see that the breakdown of the average energy per tag is given by:

$$\bar{E}_{tx} = \frac{P_{tx}}{N_t} N_t T_{tx} = P_{tx} T_{tx} \quad (4)$$

$$\bar{E}_{rx} = \frac{P_{rx}}{N_t} (n_c T_{req} + N_t T_{rx}) = P_{rx} (\eta_c T_{req} + T_{rx}) \quad (5)$$

$$\bar{E}_{cg} = \frac{P_{tx}}{N_t} (N_t T_{req} + 2n_c T_{cg}) = P_{tx} (T_{req} + 2\eta_c T_{cg}) \quad (6)$$

Where P_{tx} and P_{rx} are the power consumption of the active radios in transmit and receive mode respectively. T_{tx} is the duration of the interrogation message while T_{rx} is the time the interrogator spends listening for the reply.

2) *Latency overhead*: To describe the latency overhead we define *duration ratio* ($\eta_d = d_t/2N_t$) as the fraction of the total number of cycles needed to interrogate all tags ($d_t/2$) relative to the number needed for sequential interrogation (N_t slots). η_d relates to the latency overhead.

Transmission latency is the time that passes between the instant when the MAC layer receives the frame from the application layer until it is actually transmitted, which happens when the first suitable timeslot comes along. Therefore transmission delay is uniformly distributed between 0 and $d_s T_s$ with probability $\frac{1}{d_s T_s}$ where $d_s = d_a + d_t$ is the total number of

timeslots in the slotframe, and d_a and d_t represent the length of the regular schedule and the tags' schedule respectively. If we remember that $d_t = 2\eta_d N_t$, the average transmission latency is given by:

$$\bar{\Delta}t_{tx} = \int_0^{d_s T_s} \frac{x}{d_s T_s} dx = \frac{d_s T_s}{2} = \frac{T_s}{2} (d_a + 2\eta_d N_t). \quad (7)$$

The CDF of the delay is given by:

$$p(\Delta t_{tx} \leq x) = \begin{cases} 0 & x \leq 0 \\ \frac{x}{T_s(2\eta_d N_t + d_a)} & 0 < x \leq T_s d_s \\ 1 & x > T_s d_s \end{cases} \quad (8)$$

V. IMPLEMENTATION

To demonstrate the feasibility of our design and to support part of our experimental evaluation, we instantiate it in a network of Zolertia Firefly sensor nodes with a TI CC2538 SoC that is compatible with the IEEE 802.15.4 standard. The active nodes run the Contiki operating system [15]. As MAC protocol we employ Time-Slotted Channel Hopping (TSCH) [16], part of the IEEE 802.15.4 standard, and use RPL [17] as routing protocol.

We employ a modified version of TSCH that uses the default slot duration ($T_s = 10$ ms) and provides the necessary facilities to generate unmodulated carriers at the appropriate frequency offset and to send carrier requests when interrogating sensor tags, following the design of Section IV.

The active schedule that regular nodes employ for traffic among themselves is not part of this work. In our case it is pre-defined statically to a single time slot ($d_a = 1$) for simplicity. This schedule is used, for example, during bootstrap to collect topology information and to disseminate the tags' schedule.

We employ sensor tag prototypes with an IEEE 802.15.4 transceiver that operates assisted by an external unmodulated carrier [4]. This tag has all the characteristics described in Section II and requires a difference of 8 MHz between the unmodulated carrier and the signal of interest. They are configured so that with an unmodulated carrier strength of -70 dBm, they will have a communication range of around 25 cm. We set the minimum acceptable carrier signal strength in our algorithm to this value ($w_{min} = -70$ dBm).

To discover the topology of the active node network, we leverage the neighbor discovery information that RPL collects to establish its routes. In our implementation, during topology discovery each node periodically sends its list of neighbors to the cloud server, where it is compiled to generate a full picture of the network. RPL maintains reliable link quality information to each neighbor over time by filtering RSSI values with an exponential window moving average filter that makes it robust and adaptive to time varying link states. We employ these to construct the weighted adjacency matrix W . Note that a full-fledged topology discovery mechanism is beyond the scope of this work and there are many alternatives already in the literature [18]–[21]. Once a schedule to interrogate tags is computed, it is disseminated to all nodes and it is appended to the active nodes' schedule as additional time slots.

In place of the cloud server we employ a desktop computer with an Intel Core i7 CPU at 3.6 GHz and 16 GB of RAM running the Ubuntu operating system.

VI. EVALUATION

Our evaluation consists of two major parts. In the first part, we deploy our implementation in a testbed to show its feasibility and to demonstrate some of the attainable savings, as well as to evaluate its reliability. In the second part, we increase the scale of our evaluation by generating large numbers of instances with random tag allocations in topologies gathered from different research testbeds. We also investigate the dependency of the schedule with the network topology. To that end, we generate a large number of random topologies where we schedule tag interrogations. We also evaluate the quality of our approximate solution providing theoretical bounds for specific cases.

We make the following key findings:

- We limit the excess latency added by the inclusion of tags in the network and provide significant energy savings in accordance with the model, without affecting the reliability of tag interrogations.
- Using testbed topologies, we show that we can reduce latency by a factor of up to $N_a/2$ and there is a comparable reduction in energy consumption relative to the sequential schedule.
- Our approximate algorithm provides solutions close to optimal, even as the network grows.
- We show that the attainable savings are related to the tag distribution when it comes to latency, while the energy savings are related to the density of the network.

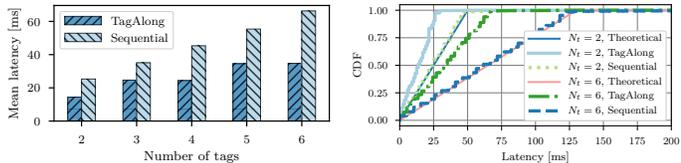
A. Experimental Evaluation

We deploy our implementation as described in Section V in a local sensor network testbed in an office building. The testbed consists of 25 active nodes that we augment with six sensor tag prototypes. Active nodes and tags have 3 dBi antennas. The active nodes transmit messages and unmodulated carriers with a nominal output power of 7 dBm. In the deployment one node has a connection to the cloud infrastructure (remote desktop computer in our case) over the open Internet where we compute the schedule. The remaining 24 nodes may act as hosts for sensor tags.

1) *Latency*: In our first experiment, we compare how the latency of communications is affected by the addition of the tag timeslots in our schedule versus the sequential one.

Setup. We configure the system in our testbed to interrogate different sets of tags. With the tag interrogation schedule in place, we transmit 1000 frames from the active nodes. We repeat this with our schedule and then with the sequential schedule, for each set of tags. We measure the transmission latency for active nodes.

Results. Figure 7 compares the transmission latency between our schedule and the sequential one. Our results confirm our theoretical predictions that the addition of tag interrogation slots increases the latency of communications linearly with



(a) Mean latency among active nodes. (b) Cumulative distribution of latency.

Fig. 7. Our schedule limits communication latency for the active nodes as we add more tags. Adding more tags increases the length of the sequential schedule, leading to increased latency. Our schedule scales much better as tags are added. Theoretical CDFs from Equation 8.

the number of tags for the sequential schedule. The results of Figure 7(a) are closely in line with the expected values according to Equation 7. Our schedule scales much better than the sequential one due to its ability to parallelize interrogations. The CDF in Figure 7(b) shows that the latency varies uniformly up to the slotframe duration and that the experimentally observed latency corresponds closely with the theoretical prediction of Equation 8.

2) *Energy Consumption:* We now examine the savings in carrier energy consumption that we obtain with our schedule.

Setup. We repeat the setup of the previous experiment to compare the sequential schedule with ours. We employ Energest [22], the energy estimation mechanism built into the Contiki operating system. In each experiment we estimate the energy the nodes spend in transmission, reception or generating carriers during the tag interrogation schedule. Note that the savings depends on the specific topology and tag placement because carrier sharing and parallelization opportunities depend on them.

Results. Figure 8 shows a comparison of the average energy invested by the active nodes per tag and per interrogation cycle for each of the radio functions. The results of Figure 8(a) show some savings in energy spent in reception due to a reduced need for carriers to listen for carrier requests. Figure 8(b) shows that the nodes invest roughly the same amount of energy for transmissions as the sequential schedule and that, in agreement with Equation 4, it does not depend on N_t ; its value is small due to the short length of interrogation frames. As a comparison, active nodes spend roughly 147 μ J to transmit a full-length IEEE 802.15.4 frame. In Figure 8(c) we can see that the sequential schedule invests roughly twice the full-frame transmission energy to interrogate one tag, which corresponds to the two carrier generation intervals.

3) *Reliability of Tag Interrogations:* To show that the reliability of interrogations does not degrade with the use of our schedule, even with synchronous interrogations, we compare the rate of success of interrogations (a request-reply cycle) when using our schedule versus interrogating tags sequentially. Figure 9 compares the mean success rate over 10 runs of 100 interrogations for each case. Error bars represent the standard deviation. The figure shows that there is very little difference in the reliability, compared to the one achieved with the sequential schedule. This is despite the fact that in

TABLE I
Evaluation Topology Details. NUMBER OF REGULAR NODES (N_a), AVERAGE NODE DEGREE AND MEAN VALUES OF CARRIER AND DURATION RATIOS FOR THE TOPOLOGIES USED IN THE EVALUATION.

Topology	N_a	Average Node Degree	Mean η_c		Mean η_d	
			$\frac{N_t}{N_a} = 0.4$	$\frac{N_t}{N_a} = 2.0$	$\frac{N_t}{N_a} = 0.4$	$\frac{N_t}{N_a} = 2.0$
Local	25	9.6	0.65	0.47	0.48	0.35
FlockLab	27	8.4	0.66	0.49	0.36	0.25
D-Cube	39	10.8	0.47	0.41	0.23	0.16

the sequential schedule there is no need for compromise in the strength of the unmodulated carrier as we can always select the best neighbor as carrier generator. We assume that the low reliability obtained for some tags, which does not depend on whether we use our schedule, is caused by external interference, as the testbed is located in an office building.

B. Schedule Evaluation with Testbed Topologies

Through our implementation we have shown that our approach is feasible and that it can provide significant savings in terms of energy and excess latency depending on η_c and η_d , as expressed in Equations 4, 5, 6 and 7. Both metrics depend on the specific topology and tag deployment. In order to perform a larger scale evaluation of our system, we conduct a set of offline experiments. We collect the topologies of the network of active nodes from our testbed and from two other open research testbeds: FlockLab [23], and D-Cube [24]. Table I shows the number of nodes in each testbed and the average node degree as a measure of the density of the networks. We employ these three topologies as the basis for our experiments, where we create random tag deployments.

Setup. For each topology we generate one hundred random tag-to-host assignments of a varying number of tags (N_t), resulting in varying average tag densities (N_t/N_a), and compute the corresponding schedule. For each instance, we compute η_d and η_c . We add up to $N_t = 78$ tags in the case of D-Cube.

Results. Figure 10 shows the average behavior of both η_d and η_c as the tag density increases, for each of the three topologies. The error bars represent the standard deviation. The figures show that both metrics improve as the density increases. We attribute this to larger tag density allowing more opportunities for carrier reuse.

C. Bounds and Dependency on Topology

To provide bounds for the value of η_d we observe that it depends on the way tags are distributed on the network. If the distribution is skewed, with all tags hosted by a single node, we will necessarily require N_t slots to interrogate them, resulting in $\eta_d = 1$. If, on the contrary, tags are uniformly distributed among the nodes, with every node hosting N_t/N_a tags, the optimal schedule will require $d_t = 2N_t/N_a$ cycles to complete. This means that $\eta_d^{opt} = \frac{d_t}{N_t} = \frac{2N_t}{N_a} \frac{1}{N_t} = \frac{2}{N_a}$. Figure 13 shows a comparison of our results with this absolute lower bound. We obtain the optimal solutions by implementing our scheduling problem in a discrete optimization solver [25]

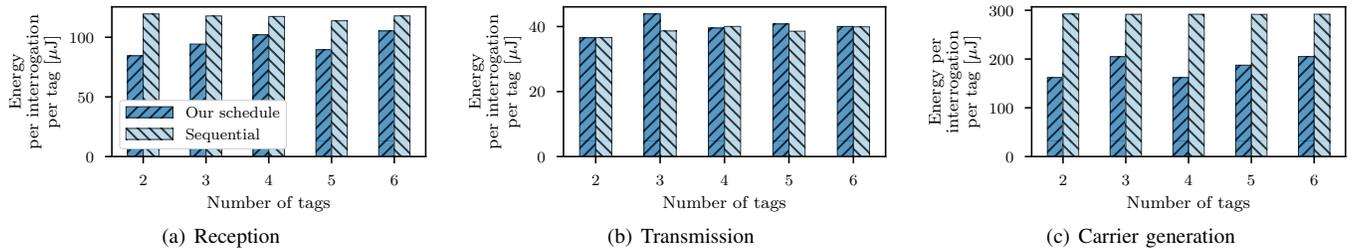


Fig. 8. Most of the energy savings are achieved due to a reduced need for carrier generation. Carrier generating nodes spend nearly twice the energy necessary to transmit a full-length frame.

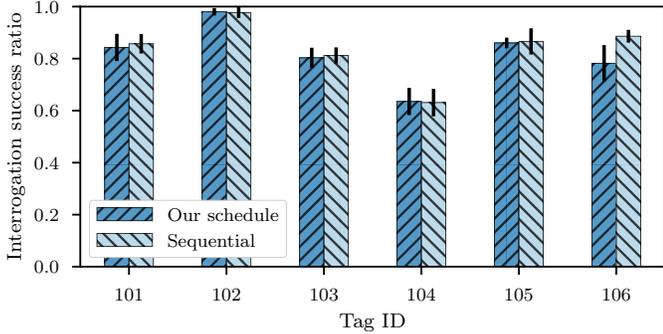


Fig. 9. There is no significant reliability penalty for using our schedule. Mean interrogation success rate over 10 runs of 100 interrogations each, error bars show the standard deviation.

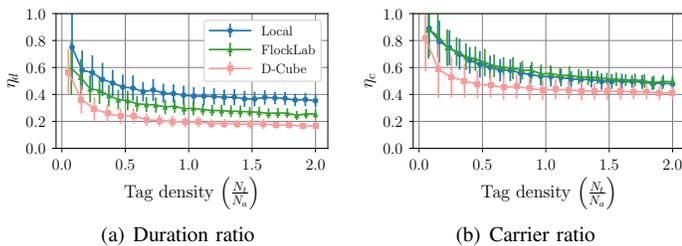


Fig. 10. Our schedule significantly improves the efficiency of tag interrogations. In average, improvements tend to increase as the number of active nodes and tags increases. Mean values over 100 tag assignments for every tested and N_t/N_a combination.

that guarantees the optimality of the solution. The figure shows how the optimal solution achieves the lower bound for uniform tag distributions but performs slightly worse for random tag assignments. Our approximate solutions are close to optimal as attested by both plots.

In the case of η_c , the theoretical lower bound is also $\eta_c^{opt} = \frac{2}{N_a}$ for uniform tag distributions, but it is only achievable in topologies with at least one node connected to all others (like a star), which is rare. Our numerical results indicate that, on average, η_c remains away from this bound and has a strong dependency with the node degree of the network but not on its size (N_a). This makes sense given that more edges per node means a higher likelihood of sharing carriers. Figure 11(b) illustrates the dependency of η_c with the number of active nodes in networks with various node degrees and random tag distributions. We can clearly see that the values improve, approaching the theoretical bound, as the average node degree increases. The values saturate quickly as

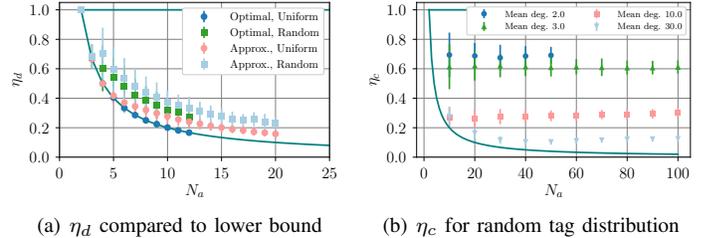


Fig. 11. η_d remains close to the absolute lower bound as the network grows, independent of the network density. η_c shows a strong dependency with network density (mean node degree).

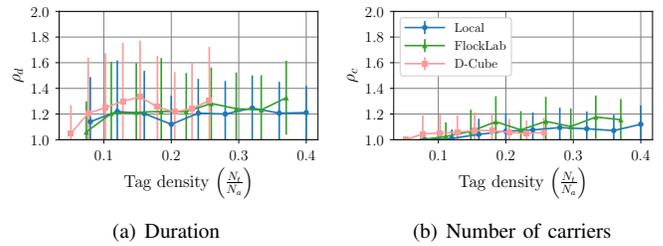


Fig. 12. Our solutions are close to the optimal even as the tag density increases. Approximation ratios suggest good scaling properties for our solution, independently of topology.

the tag density ($\frac{N_t}{N_a}$) (Figure 10(b)) and network size (N_a) (Figure 11(b)) grow.

D. Quality of Approximation

Given our approximate scheduling algorithm, we evaluate the quality of the solutions by comparing to the corresponding exact solutions. We define two approximation ratios: $\rho_d = \eta_d/\eta_d^{opt}$ and $\rho_c = \eta_c/\eta_c^{opt}$ are the ratios of our two metrics computed by our approximate algorithm to the corresponding optimal value.

Setup. We schedule 100 instances of increasing size (tag density) as in the previous section until they take more than two hours to compute. We then compare to the solutions provided by our algorithm.

Results. Figure 12 shows that our algorithm achieves good approximations, both for schedule duration and for the number of necessary carriers generators. The results also show only moderate growth in the approximation ratios as the tag density increases, suggesting that our solution should scale well to larger densities. Finally, the behaviour seems to be independent of the topology.

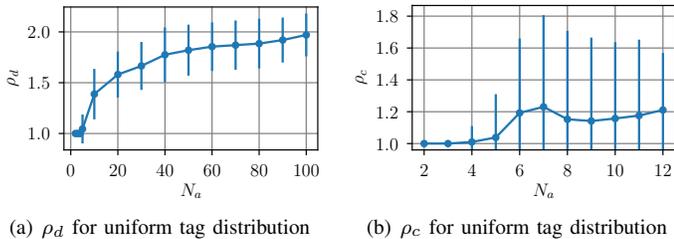


Fig. 13. The approximation ratios increase sub-linearly as the number of nodes increases for the uniform tag distribution case.

To study the behaviour of ρ_d without the limitation imposed by the computation time of the optimal solution, we evaluate the approximation ratio for the uniform tag distribution case. In the uniform case we know the optimal solution is $\eta_d^{opt} = 2/N_a$. Figure 13(a) shows that the approximation ratio increases sub-linearly with the size of the network. Because we do not have a reliable analytical lower bound for ρ_c , we can only compare with small instances solvable with the discrete optimization solver. Figure 13(b) shows that in this case ρ_c also grows slowly with the number of nodes.

VII. RELATED WORK

Our work is related to those that employ backscatter and related techniques for ultra-low power consumption, in particular to those who integrate battery-free communications with commodity networks. While the efficient provision of the unmodulated carrier is rarely addressed in the literature, in our work, we focus on providing the carrier in the most efficient way, causing as little disruption as possible to the active nodes and other nearby devices.

Our previous work introduced the idea of augmenting an IoT network with sensor tags, and having the regular nodes provide the unmodulated carrier [2], [4] but did not address carrier scheduling mechanisms. These works demonstrate the concept with a single prototype, without optimizing carrier generators in any way. We build on these works but our focus is on an efficient scheduling algorithm to provide the unmodulated carrier in an efficient way. We also demonstrate the system in a testbed with up to six sensor tag prototypes.

Netscatter [12] integrates battery-free devices into commodity networks and leverages specific properties of LoRa's physical layer to decode multiple concurrent transmissions at a specialized base station. Our work differs in that it does not assume a specific physical layer, and in that it extends a mesh network of standard IoT devices with battery-free tags.

Several other works integrate battery-free devices in standard networks [1], [5], [6], [9], [10]. These works make varying efforts to make efficient use of the unmodulated carrier. In some cases the carrier is always on, while in others its duration is tailored to the known duration of backscatter transmissions. In none of these cases, however, the authors share carriers between devices or limit the disruption of standard networks like in the way our work does.

A different approach is taken by works that employ modulated traffic as the excitation signal [8], [11], [26], [27] (ambient backscatter). While these approaches leverage an information-carrying signal as the excitation, they necessarily have less control over the exciter. As a result these works either make no attempt to optimize the exciter or sometimes try to make the active network perform spurious transmissions.

Braidio [28] takes a radical new approach by dynamically switching carrier generation between the transmitter and the receiver. Braidio focuses on maximizing the system lifetime but does not attempt to optimize carrier usage other than within a single radio link.

Finally, Van Huynh et al. [29] employ numerical analysis to optimize the overall network throughput in a network of backscatter devices powered through RF. While Van Huynh et al. focus on optimizing the energy harvesting of their tags, we do not assume, or rule out, any harvesting modality. Decoupling energy harvesting from communications allows us to directly interoperate with the standard networks while remaining independent of the harvesting modality.

Our work is similar to those looking to network battery-free devices [30]–[32]. These works concentrate on backscatter based tag-to-tag communication, by contrast we network battery-free devices over a single hop to its host. In our scenario, this more powerful host can act as a backhaul for its associated tags.

Our problem is similar to The Reader Collision Problem in RFID systems [33]–[35] in the sense that both need to avoid collisions from the carrier. These works are different from ours in that they focus on the monostatic backscatter configuration where the carrier generator and receiver are co-located whereas our work focuses on the bi-static configuration, i.e. separated carrier generators and receivers. The bi-static configuration leads to a different, previously unexplored, optimization problem and our focus goes beyond avoiding collisions to also optimizing resources.

VIII. CONCLUSIONS

In this paper we considered the problem of scheduling interrogations of battery-free sensor tags that interoperate with a network of regular IoT nodes. We proposed an efficient approximate algorithm with time complexity $\mathcal{O}(N_t N_a^2)$ and, through our testbed-based evaluation, we have shown that our scheme is feasible. We have shown that our schedule can significantly reduce the disruption to the operation of active nodes in terms of added latency, energy consumption and spectrum usage up to a factor $N_a/2$; all this with no discernible decrease in reliability when compared to the alternative approach of interrogating each tag in sequence. We showed that our solutions are, on average, close to the optimal and provided important insights on the relationship between the achievable gains and the underlying network topology.

ACKNOWLEDGEMENTS

This work has been partly funded by the Swedish Research Council (Grants 2017-045989 and 2018-05480).

REFERENCES

- [1] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, "Passive Wi-Fi: Bringing Low Power to Wi-Fi Transmissions," ser. NSDI '16, 2016, pp. 151–164.
- [2] C. Pérez-Penichet, F. Hermans, A. Varshney, and T. Voigt, "Augmenting IoT Networks with Backscatter-enabled Passive Sensor Tags," in *Proceedings of the 3rd Workshop on Hot Topics in Wireless*, ser. HotWireless '16. New York, NY, USA: ACM, 2016, pp. 23–27. [Online]. Available: <http://doi.acm.org/10.1145/2980115.2980132>
- [3] J. Ensworth, A. Hoang, and M. S. Reynolds, "A low power 2.4 GHz superheterodyne receiver architecture with external LO for wirelessly powered backscatter tags and sensors," in *2017 IEEE International Conference on RFID (IEEE RFID)*, 2017, pp. 149–154.
- [4] C. Pérez-Penichet, C. Noda, A. Varshney, and T. Voigt, "Battery-free 802.15.4 Receiver," in *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '18, 2018, pp. 164–175. [Online]. Available: <https://doi.org/10.1109/IPSNS.2018.00045>
- [5] A. Varshney, O. Harms, C. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt, "LoRea: A Backscatter architecture that achieves a long communication range," in *ACM SenSys 2017*. ACM Digital Library, 2017.
- [6] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-Technology Backscatter: Towards Internet Connectivity for Implanted Devices," in *Proceedings of the ACM SIGCOMM Conference*, ser. SIGCOMM '16. ACM, 2016, pp. 356–369. [Online]. Available: <http://doi.acm.org/10.1145/2934872.2934894>
- [7] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient Backscatter: Wireless Communication out of Thin Air," in *Proceedings of the ACM SIGCOMM Conference*, ser. SIGCOMM '13. ACM, 2013, pp. 39–50. [Online]. Available: <http://doi.acm.org/10.1145/2486001.2486015>
- [8] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "HitchHike: Practical Backscatter Using Commodity WiFi," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '16. ACM, 2016, pp. 259–271. [Online]. Available: <http://doi.acm.org/10.1145/2994551.2994565>
- [9] J. Ensworth and M. S. Reynolds, "Every smart phone is a backscatter reader: Modulated backscatter compatibility with Bluetooth 4.0 Low Energy (BLE) devices," in *IEEE RFID 2015*, 2015.
- [10] V. Talla, M. Hesar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity," vol. 1, no. 3, Sep. 2017, pp. 105:1–105:24. [Online]. Available: <http://doi.acm.org/10.1145/3130970>
- [11] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "Plora: A passive long-range data network from ambient lora transmissions," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '18. New York, NY, USA: ACM, 2018, pp. 147–160. [Online]. Available: <http://doi.acm.org/10.1145/3230543.3230567>
- [12] M. Hesar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *NSDI 2018*, 2018.
- [13] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. Wiley-Interscience, 2005.
- [14] A. Kosowski and K. Manuszewski, "Classical coloring of graphs," *Contemporary Mathematics*, vol. 352, pp. 1–20, 2004.
- [15] A. Dunkels, T. Voigt, and B. Grönvall, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceedings. 29th Annual IEEE International Conference on Local Computer Networks. LCN 2004*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2004, pp. 455–462. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/LCN.2004.38>
- [16] IEEE, *IEEE Standard for Low-Rate Wireless Networks*, Apr. 2015.
- [17] T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks." [Online]. Available: <https://tools.ietf.org/html/rfc6550>
- [18] N. Karowski, K. Miller, and A. Wolisz, "Greedy multi-channel neighbor discovery," *CoRR*, vol. abs/1807.05220, 2018.
- [19] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noel, "Low-power neighbor discovery for mobility-aware wireless sensor networks," *Ad Hoc Netw.*, vol. 48, no. C, pp. 66–79, Sep. 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.adhoc.2016.05.011>
- [20] M. Seliem, K. Elsayed, and A. Khattab, "Performance evaluation and optimization of neighbor discovery implementation over Contiki OS," 2014, pp. 119–123.
- [21] S. Devasenapathy, V. Rao, V. Prasad, I. Niemegeers, and A. R. Biswas, "Between neighbors: Neighbor discovery analysis in EH-IoTs," in *10th International Conference on Autonomic Computing (ICAC '13)*, 2013.
- [22] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th Workshop on Embedded Networked Sensors*, ser. EmNets '07. New York, NY, USA: ACM, 2007, pp. 28–32. [Online]. Available: <http://doi.acm.org/10.1145/1278972.1278979>
- [23] R. Lim, F. Ferrari, M. Zimmerling, C. Walsler, P. Sommer, and J. Beutel, "Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, ser. IPSN '13. New York, NY, USA: ACM, 2013, pp. 153–166. [Online]. Available: <http://doi.acm.org/10.1145/2461381.2461402>
- [24] M. Schuß, C. A. Boano, M. Weber, and K. Römer, "A competition to push the dependability of low-power wireless protocols to the edge," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '17. USA: Junction Publishing, 2017, pp. 54–65. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3108009.3108018>
- [25] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack, "MiniZinc: Towards a Standard CP Modelling Language," in *Principles and Practice of Constraint Programming – CP 2007*, ser. Lecture Notes in Computer Science, C. Bessière, Ed. Springer Berlin Heidelberg, 2007, pp. 529–543.
- [26] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-Fi Backscatter: Internet Connectivity for RF-powered Devices," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 607–618. [Online]. Available: <http://doi.acm.org/10.1145/2619239.2626319>
- [27] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "FreeRider: Backscatter Communication Using Commodity Radios," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17. Incheon, Republic of Korea: ACM, 2017, pp. 389–401. [Online]. Available: <http://doi.acm.org/10.1145/3143361.3143374>
- [28] P. Hu, P. Zhang, M. Rostami, and D. Ganesan, "Braidio: An Integrated Active-Passive Radio for Mobile Devices with Asymmetric Energy Budgets," in *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference*, ser. SIGCOMM '16. New York, NY, USA: ACM, 2016, pp. 384–397. [Online]. Available: <http://doi.acm.org/10.1145/2934872.2934902>
- [29] N. V. Huynh, D. T. Hoang, D. Niyato, P. Wang, and D. I. Kim, "Optimal time scheduling for wireless-powered backscatter communication networks," *IEEE Wireless Communications Letters*, vol. 7, pp. 820–823, 2018.
- [30] A. Y. Majid, M. Jansen, G. O. Delgado, K. S. Yildirim, and P. Pawełzak, "Multi-hop backscatter tag-to-tag networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, April 2019, pp. 721–729.
- [31] P. V. Nikitin, S. Ramamurthy, R. Martinez, and K. V. S. Rao, "Passive tag-to-tag communication," in *2012 IEEE International Conference on RFID (IEEE RFID)*, April 2012, pp. 177–184.
- [32] Y. Karimi, A. Athalye, S. R. Das, P. M. Djurić, and M. Stanačević, "Design of a backscatter-based tag-to-tag system," in *2017 IEEE International Conference on RFID (IEEE RFID)*, May 2017, pp. 6–12.
- [33] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, "Season: Shelving interference and joint identification in large-scale RFID systems," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 3092–3100.
- [34] E. Hamouda, N. Mitton, and D. Simplot-Ryl, "Reader Anti-collision in dense RFID networks with mobile tags," in *2011 IEEE International Conference on RFID-Technologies and Applications*, Sep. 2011, pp. 327–334.
- [35] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A time-efficient information collection protocol for large-scale RFID systems," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 2158–2166.